Application of Face Detection and Recognition Algorithm Based on Deep Learning in Automatic Interception of Target Person Video Clips

Zhibiao Wang, Ye Tao*, Xu Liu, Jiayi Yu, and Wenhua Cui

Abstract—**Current automatic segment extraction techniques for identifying target characters in videos have several limitations, including low accuracy, slow processing speeds, and poor adaptability to diverse scenes. This paper introduces an optimized algorithm to address these issues that enhance the RetinaFace and FaceNet models. We selected RetinaFace for face detection, employing MobileNetV1-0.25 as the backbone network and simplifying its Feature Pyramid Network (FPN) structure to boost detection speeds. Analysis of 460 images with a 720P resolution demonstrated an average speed improvement of 20.6%. For face recognition, we utilized FaceNet with MobileNetV3 as the backbone, augmenting its feature extraction capability by integrating four Receptive Field Block (RFB) structures and replacing the Squeeze-and-Excitation (SE) module with the Convolutional Block Attention Module (CBAM). Experimental results indicate that our enhancements elevate the maximum accuracy to 97%, outperforming the original model. Additionally, we integrated these refined algorithms and conducted disintegration experiments on segment extraction in 10 videos, evaluating various metrics. The findings show improvements in both precision and recall. We also compared our algorithm against the Dlib model; our system achieved an overall interception accuracy of 79.94%, surpassing Dlib's 75.55%. This confirms the enhanced performance and feasibility of our proposed algorithm.**

Index Terms—**Face detection, Face recognition, Character interception, RetinaFace, FaceNet**

Manuscript received November 20, 2023; revised July 4, 2024.

This work was supported by Joint Fund Project of the National Natural Science Foundation of China (U1908218), the Natural Science Foundation project of Liaoning Province (2021-KF-12-06), and the Department of Education of Liaoning Province (LJKFZ20220197).

Zhibiao Wang is an undergraduate student of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China. (e-mail: wzb021120@163.com).

Ye Tao is an Associate Professor of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China. (Corresponding author to provide phone: +86-133-0422-4928; e-mail: taibeijack@163.com).

Xu Liu is a graduate student of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China. (e-mail: 1441813628@qq.com).

Jiayi Yu is an undergraduate student of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China. (e-mail: 1551009136@qq.com).

Wenhua Cui is a Professor of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, China. (e-mail: taibeijack@126.com).

I. Introduction

With the onset of the 5G era, the volume and importance of video data have surged dramatically. Video data has become increasingly crucial because of its ease of collection, durability, and retention of comprehensive information [1]. Unlike conventional big data, video data adds extra dimensions, generates larger volumes, and is inherently unstructured, presenting significant challenges for processing.

The burgeoning industry of short videos [2], which has grown significantly in recent years, relies heavily on video editing technologies. These technologies are crucial across various fields, including cinema, animation, and web-based video platforms. The recent advancements in deep learning have catalyzed significant developments in the automation of video data processing. One such development is the technology for automatic extraction of target character segments from video streams. This technology facilitates the automated editing of specific video segments, highlighting their research significance and practical value.

Face recognition technology intersects with domains, such as digital image processing [3], computer vision, pattern recognition, and machine learning. Its applications are widespread and include face identification [4], social security measures, expression recognition [5], and feature classification [6]. Research and development in face recognition technology have progressed through three distinct phases: the initial stage, which introduced basic recognition techniques; the development stage, which enhanced reliability and accuracy; and the adaptive learning stage, which employs advanced machine learning algorithms to improve performance. These phases have introduced various methods, including Feature Invariant Approaches, Template Matching Methods, and Knowledge-based Methods, the principles of which are summarized in Table I.

Central to any face recognition system are the processes of detection and recognition. Face detection algorithms primarily aim to identify the presence of faces within an image and refine the image to emphasize facial features. The RetinaFace algorithm [7], introduced in 2019 and an enhancement of the RetinaNet, integrates the Feature Pyramid Network (FPN) and SSH structures to improve detection efficacy. As depicted in Figure 1, inputting an image into the network outlines the faces detected within.

TABLE I
INTRODUCTION OF THREE BASIC FACE RECOGNITION TECHNOLOGIES

Method classification	Realization principle and method
Feature Invariant Approaches	Through the partial or overall features of the face, such as shape, texture, skin color, outline, and other features that do not change with external factors (such as light, posture, etc.), these features are extracted through specific methods to establish a feature model of the face. Specific methods include the Haar feature [8], LBP (Local Binary Pattern) feature, SIFT feature, Gabor feature [9], HOG (Histogram of Oriented Gradient) feature, etc.
Template Matching Methods	Standardize and parameterize the facial features of the face to form a face feature template and perform face detection by calculating the correlation value between the image to be detected and the face template. The specific methods are the active appearance model (Active Appearance Models, AAM) and the active shape model (Active Shape Models, ASM).
Knowledge-based Methods	Construct the inherent feature rules of many face images, and use the rules for matching to realize face recognition and detection.

Fig. 1. Detection effect diagram of the RetinaFace algorithm

Challenges in traditional face recognition, such as varying poses and lighting conditions [10], have been addressed by the FaceNet algorithm, which normalizes features and maps them to a Euclidean space for comparison. Figure 2 depicts the Euclidean distance [11] serves as a metric for distinguishing between individuals, with a threshold value set at 1.1. If the Euclidean distance between two images exceeds 1.1, it is considered that the images belong to different individuals. Conversely, if the Euclidean distance is less than 1.1, it is concluded that the two images represent the same person.

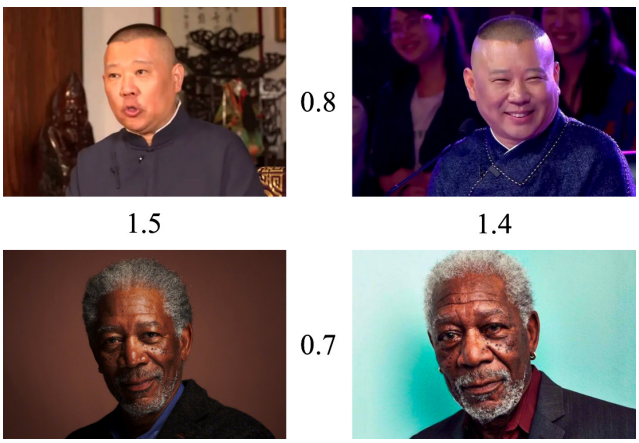


Fig. 2. Euclidean distance comparison

In 2015, Google introduced the FaceNet face recognition algorithm [12], which has since become a widely used network in this field. The initial version of FaceNet utilized two types of deep convolutional networks. The first is based on the Zeiler & Fergus model [13], introduced in 2013, which is an evolution of the AlexNet architecture. The second is based on Google's Inceptionv1 model, also known as GoogLeNet, which debuted in 2014.

In their studies, Zhenyao [14] and colleagues utilized deep networks to normalize distorted facial images, which were then incorporated into a convolutional neural network (CNN) containing known identity data. This approach integrates Support Vector Machines (SVM) and Principal Component Analysis (PCA) for robust face verification. Taigman [15] expanded on this by employing a multi-stage research methodology. The initial facial input is aligned with a standard 3D model and subsequently assessed using a multi-class network model capable of recognizing over 4,000 identities. Furthermore, experiments were conducted using a twin network model to refine the L1 distance measurement between facial features.

The initial implementation of the FaceNet algorithm employed the Inception-ResNet network model as its primary feature extraction network. This model combines elements from Google's Inception networks [16]-[18] and Microsoft's concept of residual networks, enhancing depth while significantly reducing the number of parameters. Research focuses on enhancing efficiency and simplifying the network architecture, spurred by recent advancements in dynamic quantitative network models that offer fresh perspectives for FaceNet's innovation.

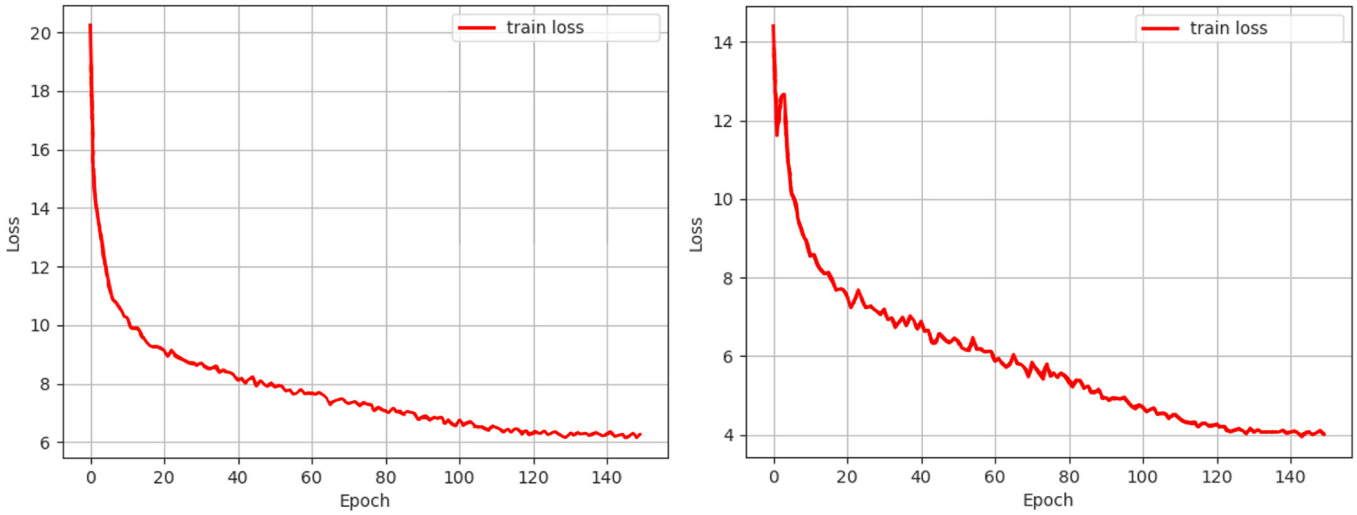
However, the RetinaFace algorithm faces challenges, notably its prolonged image processing time. It may hinder its application in real-time scenarios and fail to meet the demands of large-scale data sets or rapid processing needs. Similarly, FaceNet struggles with a substantial backbone network that consumes considerable computing resources and memory and may need more feature extraction capabilities in complex scenarios such as variable lighting, angles, or partial occlusions. This limitation often restricts the model to local feature extraction, which can degrade the accuracy of face recognition.

In response to these challenges, this paper proposes optimizations to the RetinaFace and FaceNet algorithms, introducing an automatic segment extraction system for identifying specific individuals in videos. This enhanced system aims to provide more precise and quicker extraction of targeted video segments, improving performance and applicability.

II. IMPROVED FACE DETECTION ALGORITHM BASED ON RETINAFACE

A. Selection of the Backbone Network

The RetinaFace algorithm supports various backbone networks, with MobileNet and ResNet [19] being the most commonly used. This paper presents two network models, one utilizing MobileNetV1-0.25 and the other using ResNet. The performance of the RetinaFace algorithm, trained with these backbone networks, is analyzed, and the associated loss functions are illustrated in Figure 3.



a) MobileNetV1-0.25 b) ResNet

Fig. 3. Loss function

Table II compares the results of the RetinaFace face detection algorithm using these two backbones. While the model based on MobileNetV1-0.25 demonstrates slightly lower detection accuracy compared to the ResNet-based model, it has a significantly smaller model size—just 1/50th of the ResNet model. Consequently, this paper selects MobileNetV1-0.25 as the backbone feature extraction network for RetinaFace.

TABLE II
UNITS FOR MAGNETIC PROPERTIES

Main network	Model size	Enter image size	Easy	Medium	Hard
MobileNet V1-0.25	2Mb	1280×1280	89.72%	86.71%	73.27%
ResNet	105Mb	1280×1280	94.69%	93.08%	84.31%

Using MobileNetV1-0.25 as the backbone extracts three prominent feature layers: C3, C4, and C5. These layers serve as inputs for the subsequent parts of the network.

B. Construction of the FPN Feature Pyramid and Enhancement of the SSH Feature Extraction Layer

After extracting three adequate feature layers—C3, C4, and C5, they are integrated using a Feature Pyramid Network (FPN). The construction of the FPN is illustrated in Figure 4. Each of the three layers initially undergoes a 1×1 convolution to adjust the number of channels. Subsequently, the smallest layer is upsampled to increase its dimensions. Once enlarged, it is combined with the second feature layer, which has also been adjusted for channel count. This combination process facilitates feature fusion.

Following this initial fusion, a 64-channel 1×1 convolution is performed to further standardize the number of channels. to standardize the number of channels further. Another upsampling step is conducted before adding the largest feature layer, again adjusted for channel count. A final 1×1 convolution with 64 channels is applied to refine the output.

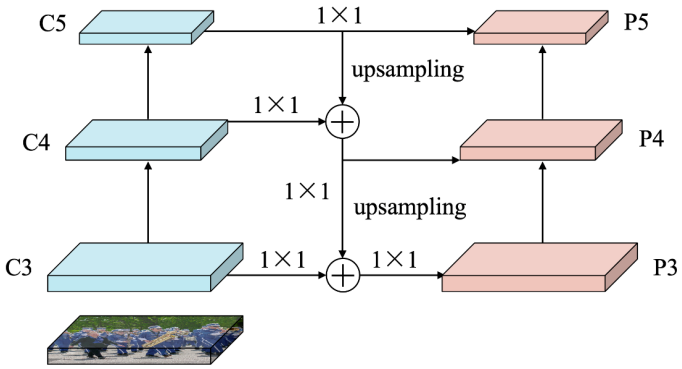


Fig. 4. Construction of FPN

As a result of the FPN operations, three initial predictive feature layers, P3, P4, and P5, are derived. These layers are then processed through the SSH (Single Stage Headless) module, which enhances the receptive field of the feature layer. Figure 5 shows the SSH module, which consists of three parallel structures. Each structure includes one 3×3 standard convolution, supplemented by two additional 3×3 convolutions simulating a 5×5 convolution effect, and three 3×3 convolutions emulating a 7×7 convolution. This configuration enhances the module's capability to handle varying feature scales effectively.

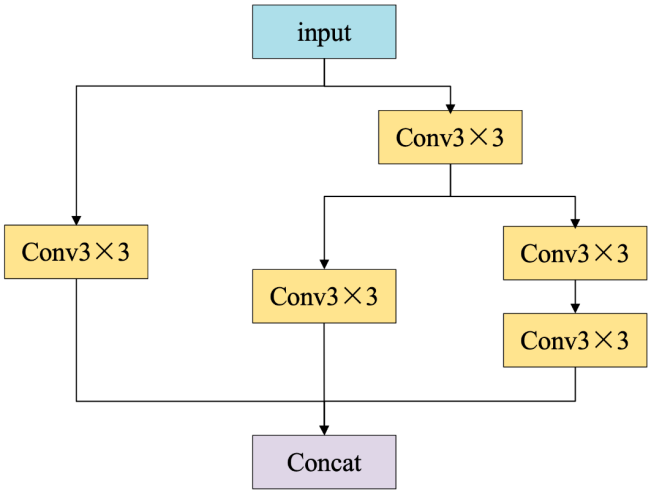


Fig. 5. Structure diagram of SSH module

C. Analysis of RetinaFace Prediction Results

Upon processing through the SSH module, the system generates three final impactful feature layers, named SSH1, SSH2, and SSH3. These layers are used to make predictions in RetinaFace, categorized into three types: face classification, regression prediction of the human face frame, and regression prediction of facial key points. The primary outcomes are detailed below:

(1) Face Classification: This process determines whether a face exists within the prior box at each grid point. A 1×1 convolution adjusts the number of input feature layer channels to num_anchors × 2. In RetinaFace, num_anchors refers to the number of prior boxes at each grid point, typically set to 2. This configuration aids in identifying the presence of a face in the prior box; a high value at serial number 1 indicates the presence of a face, whereas a high value at serial number 0 indicates its absence.

(2) Face Frame Regression Prediction: This function adjusts the dimensions and position of the prior frame to fit the detected face accurately, involving width, height, and center point coordinates. Each parameter requires four channels, so the number of input feature layer channels is adjusted using a 1×1 convolution to num_anchors × 4.

(3) Facial Keypoints Regression Prediction: The aim is to predict the coordinates for five vital facial points. The number of input feature layer channels is adjusted to num_anchors × 10 through a 1×1 convolution. The number 10 is derived from 5×2, where '5' represents the number of facial vital points, and '2' refers to the x and y coordinates needed for each point.

D. Face Detection Dataset

The face detection study utilized the Wider Face dataset, a well-known publicly available dataset shown in Figure 6. It consists of 16,106 annotated images with facial details, split into 12,880 training images and 3,226 test images [20].

Fig. 6. Picture display in the dataset

E. Adjust the FPN Feature Pyramid

Figure 7 illustrates the overall structure of RetinaFace. The architecture utilizes a backbone with a Feature Pyramid Network (FPN) for feature fusion. The ContextModule further processes each feature map to extract additional contextual information, which aids in predicting facial confidence scores, bounding box coordinates, and facial key point locations.

As demonstrated in Figure 7, the P5 feature map possesses the minor receptive field and the finest feature granularity, making it suitable for detecting small faces using smaller anchors. Conversely, the P2 feature map, with the largest receptive field and the coarsest feature granularity, is used with more prominent anchors to detect more giant faces.

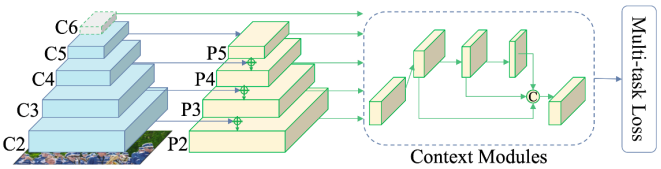


Fig. 7. Overall structure diagram of the RetinaFace

Upon analyzing the original FPN feature pyramid, we optimized RetinaFace by focusing model predictions on more significant faces and excluding results for smaller faces. This decision is based on the observation that detecting smaller faces contributes minimally to the accuracy and recall of face recognition tasks. Consequently, removing the function to detect small faces in such scenarios conserves computational resources and enhances the model's inference speed. Figure 8 depicts the revised diagram of the FPN structure, highlighting these adjustments.

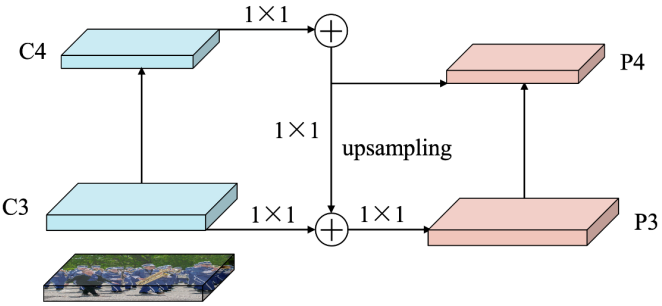


Fig. 8. Improved FPN structure diagram

F. Experimental Comparison and Result Analysis

In training the RetinaFace algorithm model, we set the learning rate to 0.01, defined the batch size as 8, and conducted 150 training iterations. After the experiment, we analyzed the processing speed of 460 images with 720p resolution. We compared the face detection speeds before and after simplifying the FPN and against classical face detection algorithms such as VGG and CNN. The results in Table III indicate that the improved RetinaFace algorithm model achieves a faster average detection speed than its previous version and the compared classical algorithms.

TABLE III
COMPARISON OF RESULTS

Algorithm	Average time spent (ms)
VGG	115
CNN	67
RetinaFace	16.5
Algorithm in this paper	13.1

a) Improved face detection

b) Face detection before improvement

Fig. 9. Comparison of detection results before and after improvement

Figure 9 demonstrates the face detection performance of the RetinaFace algorithm before and after the improvements were implemented. The primary enhancement involved eliminating the functionality for predicting small-sized face detection boxes in the initial stage of the FPN on the RetinaFace backbone. This modification leads to the exclusion of only relatively small faces from detection, which has negligible impact on subsequent processes, such as face recognition accuracy and recall.

TABLE IV
COMPARISON OF EXPERIMENTAL RESULTS BEFORE AND AFTER IMPROVEMENT

Backbone network	Easy	Medium	Hard	FPS
Resnet	94.69%	93.08%	84.31%	19.0
Resnet+FPN	93.52%	92.01%	82.12%	21.0
Mobilenet	89.72%	86.71%	73.27%	60.3
Mobilenet+FPN	88.91%	86.11%	72.12%	76.1

In practical testing, on the Mobilenet model, the face detection speed of 720p resolution images increased by 2.6ms, reaching approximately 5.47fps. As shown in Table IV, this improvement is observed on the Mobilenet and the Resnet network after the feature pyramid is enhanced. Although the detection accuracy is reduced, the downstream task is face recognition, and the decrease in accuracy is due to discarding detections of smaller faces, which has little effect on subsequent face recognition. However, the face detection speed has significantly improved after the enhancement.

III. IMPROVED FACE RECOGNITION ALGORITHM BASED ON FACENET

The face recognition process begins with the RetinaFace algorithm for detecting faces, followed by employing the FaceNet algorithm to extract facial features from the detected faces.

A. Face Image Normalization

In natural shooting conditions, it is common for people in video footage to not always face the camera directly. Also, the distance between the subject and the recording equipment can vary, causing differences in the size and orientation of the captured faces.

Most faces in video frames are not straight-on because of changes in body posture, which can make the face appear tilted. This makes face recognition more complex. To address this issue, we use image-level normalization techniques.

The normalization process begins by checking the position of the eyes in the face image to see if the face is horizontal. Usually, the line between the eyes should be horizontal if the face is aligned correctly. However, due to minor errors in the identification process, significant deviations from horizontal are standard in video data. To fix this, we carry out horizontal normalization on the face images. This involves calculating the tilt angle and rotating the image to achieve a horizontal orientation. The steps involved in this horizontal normalization process are detailed in Figure 10.

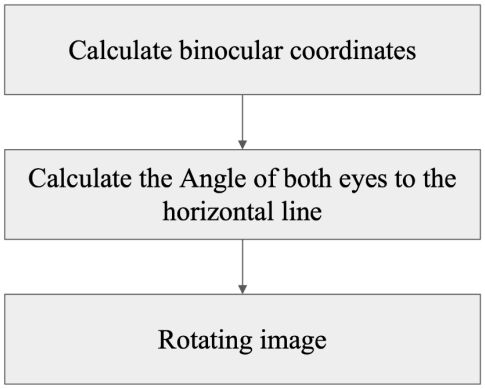


Fig. 10. The overall process of horizontal normalization

The specific steps are as follows.

(1) In the face picture obtained from the video data. Make sure the coordinates of the left eye are marked as (x_1, y_1). The coordinates of the right eye are marked (x_2, y_2).

(2) Note that the angle between the line connecting the eyes and the horizontal line is α, then the calculation method of α is:

$$\alpha = \arctan\left(\frac{y_2 - y_1}{x_2 - x_1}\right) \quad (1)$$

(3) The declination angle α between the image and the horizontal direction is obtained from the formula (3-9). Then, the image must be reversely rotated by an angle α to obtain a horizontal image. Note (x, y) as the coordinates of the original image. (x', y') is the rotated image, which can be

obtained from the rotation angle α:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} cos\ \alpha & sin\ \alpha \\ -sin\ \alpha & cos\ \alpha \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (2)$$

After horizontal normalization, the issue of the face's horizontal angle is effectively resolved, ensuring optimal horizontal alignment. This adjustment significantly reduces the impact of other factors, such as angle variations, on subsequent face detection and recognition processes.

During video recording, the uncontrollable nature of human movement means that the distance between the camera and the subjects can change constantly. Consequently, the size of the faces in the recorded images varies. To address this, size normalization becomes a crucial step in the preprocessing of video images. The specific steps involved in this process are detailed in Figure 11.

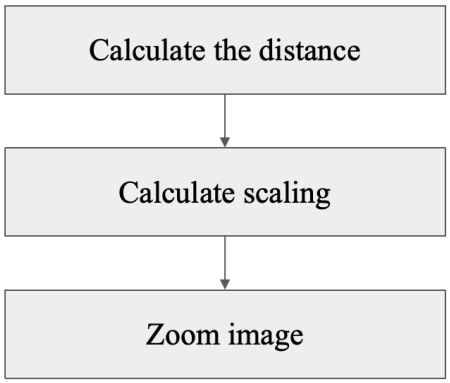


Fig. 11. The overall flow of size normalization

(1) In the obtained face image. Determine the coordinates of the left eye as (x_1, y_1) and the coordinates of the right eye as (x_2, y_2).

(2) Calculate the distance d between the eyes. d is calculated as the formula:

$$d = x_2 - x_1 \quad (3)$$

(3) Determine a standard distance d_0. Calculate the scaling ratio L corresponding to each image:

$$L = \frac{d}{d_0} \quad (4)$$

(4) Scale the entire image according to the scaling ratio of each image. For each coordinate (m, n), the scaled coordinate is (m_0, n_0). The calculation method is the formula:

$$m_0 = \frac{m}{L} \qquad n_0 = \frac{n}{L} \quad (5)$$

B. Face Recognition Dataset

Fig. 12. Partial dataset picture display

The face recognition dataset used in this study is sourced from the publicly available CASIA-WebFace dataset, as illustrated in Figure 12. For this research, we selected and aligned available faces from the dataset. The organization of the dataset involves creating multiple subdirectories within the leading dataset directory. Each subdirectory contains numerous face images, all representing the same individual.

C. Workflow of FaceNet

As illustrated in Figure 13, the FaceNet algorithm consists of three main components. The first component is the backbone feature extraction network, which extracts the initial feature from the input face image, obtaining preliminary facial features.

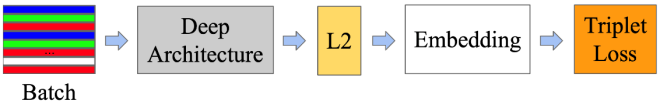


Fig. 13. FaceNet algorithm workflow

These preliminary features are processed in the second stage to produce a 128-length feature vector. This involves executing a global average pooling operation [21] on the preliminary feature layer to create an unprocessed feature strip, followed by a fully connected layer with 128 neurons. This process yields an initial feature vector of length 128.

The third component involves performing L2 normalization on this 128-length feature vector. The rationale behind L2 normalization is to standardize the magnitudes of the feature strips obtained from the second stage. Without this step, the feature strips could vary significantly in magnitude, leading to instability in the network during face recognition tasks. L2 normalization adjusts the feature strips to a uniform magnitude [22], ensuring that the modulus of the input face feature strips is standardized to 1. This enhances the stability of the network, facilitating more reliable face comparison operations.

In Figure 13, "Batch" refers to batches of input face images detected and cropped to a uniform size for processing. These images first undergo feature extraction and then L2 normalization. Subsequently, classification is performed using the Triplet loss function, which aims to minimize the distance between feature vectors of the same identity and maximize the distance between those of different identities.

Once the face feature vector of length 128 is normalized during face recognition, it is compared with existing vectors in the face database. Suppose a vector in the database shows a close Euclidean distance to the newly obtained vector. In that case, it indicates a high similarity between the two faces, suggesting they likely belong to the same individual.

D. Select Backbone Network

The FaceNet algorithm offers a variety of options for the backbone network, including MobileNet and Inception-ResNetV1. In our research, we constructed network models using MobileNetV1, MobileNetV3, and Inception-ResNetV1 [23] as backbones, as detailed in Table V. Our findings indicate that the model based on MobileNetV1 achieved a recognition accuracy of 96.81%. In comparison, MobileNetV3 reached 96.96%, and Inception-ResNetV1 achieved the highest accuracy at 97.41%. However, these models exhibit substantial

parameter count and size differences. Specifically, MobileNetV1 has 3,360,707 parameters, MobileNetV3 has 3,338,304, and Inception-ResNetV1 has 22,793,728, approximately seven times more than the MobileNet models. Given the impact of model size on experimental equipment selection and the minimal difference in accuracy among the models, our paper selects MobileNetV3 as the backbone feature extraction network for FaceNet.

Furthermore, the thresholds in Table V function as decision boundaries. Distances exceeding these thresholds signal different individuals, while distances below the thresholds indicate the same person.

TABLE V
EXPERIMENT COMPARISON

Model	Accuracy	Parameter	Threshold
MobileNetV1	96.81%	3360707	1.05
Inception-ResNetV1	97.41%	22793728	1.04
MobileNetV3	96.96%	3338304	1.05

The FaceNet face recognition algorithm utilizes a chosen backbone network to extract a 128-dimensional feature vector of the face. This vector is then analyzed using the Triplet Loss function, which aids in identifying similar samples by quantifying them as Euclidean distances. This method effectively verifies individual facial features by leveraging the principles of similarity, enabling accurate authentication. For a visual representation of the algorithm's structure, refer to Figure 14.

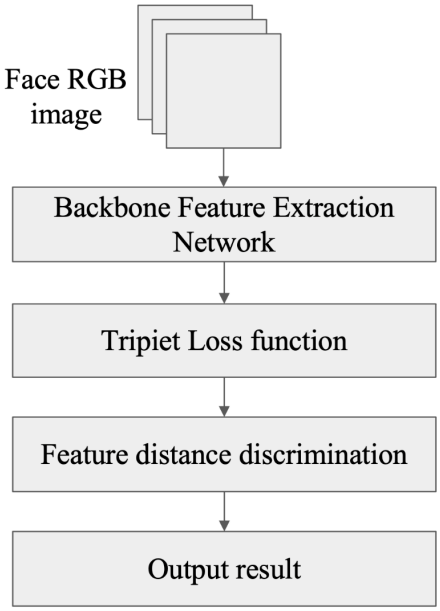


Fig. 14. FaceNet algorithm structure

Let the eigenvector input to the Euclidean space be x_i^a. The positive sample feature vector is x_i^p. The egative sample feature vector is x_i^n. Then the expression of the TripletLoss loss function is

$$\| f(x_i^a) - f(x_i^p) \|_2^2 + \alpha < \| f(x_i^a) - f(x_i^n) \|_2^2 \quad \forall \left(f(x_i^a), f(x_i^p), f(x_i^n)\right) \in T \tag{6}$$

In Equation (6), α is the positive and negative sample boundary value. T is the set of possible triples in the data set. n represents the number of elements in the set.

Triplets, central to this process, consist of an Anchor, a Negative, and a Positive image. The Anchor is the reference image, the Positive is an image from the same category as the Anchor, and the Negative is from a different category. The Triplet Loss function aims to minimize the distance between the Anchor and the Positive while maximizing the distance between the Anchor and the Negative, thereby confirming their positions in the Euclidean space. This process helps to judge the authenticity of the relationships more accurately. The operation of the Triplet Loss function is detailed in Figure 15.

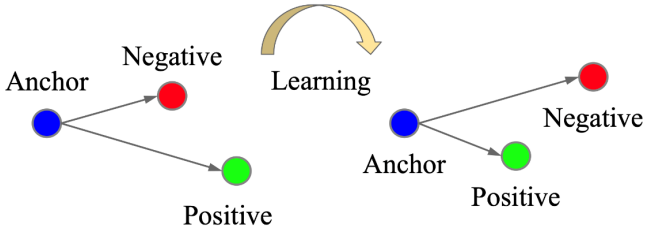


Fig. 15. The process of Triplet Loss

The similarity of corresponding samples is obtained by comparing the Euclidean distance of the feature vector x_i^a. The formula for calculating Euclidean distance is

$$D = \sqrt{\left(x_i^1 - x_j^1\right) + \left(x_i^2 - x_j^2\right) + \cdots + \left(x_i^d - x_j^d\right)} \tag{7}$$

In Equation (7), i and j represent the sample ID participating in the training. d represents the dimension. D represents the Euclidean distance between two samples.

E. Attention Module and RFB Structure

In research, we have improved the backbone network model by integrating the CBAM attention mechanism, which consists of channel and spatial attention modules, as depicted in Figure 16.

The channel attention module, illustrated in the upper section of Figure 16, operates as follows: It uses global average pooling (AvgPool) and global maximum pooling (MaxPool) on the input feature layer to capture spatial dimensions. The results of these pooling operations are then fed into a shared fully connected layer (SharedMLP). The outputs from this layer are combined and processed through a Sigmoid activation function, generating a channel attention map. This map specifies the weights assigned to each channel of the input feature layer, which is then used to enhance specific features by channel-wise multiplication with the input feature layer.

On the other hand, the spatial attention module, shown in the lower part of Figure 16, evaluates the significance of different regions within the input image, complementing the channel attention module. Spatial attention is computed by conducting average pooling and max pooling across the channel dimension of each feature point to create a stacked feature descriptor. This descriptor is then processed through a convolutional layer with adjusted channel numbers, and the output undergoes Sigmoid activation. The resulting two-dimensional spatial attention map assigns weight values to each point in the input feature map. These weights are then

applied channel-wise to the input feature layer, similar to the process in the channel attention module, refining the focus on significant spatial features.

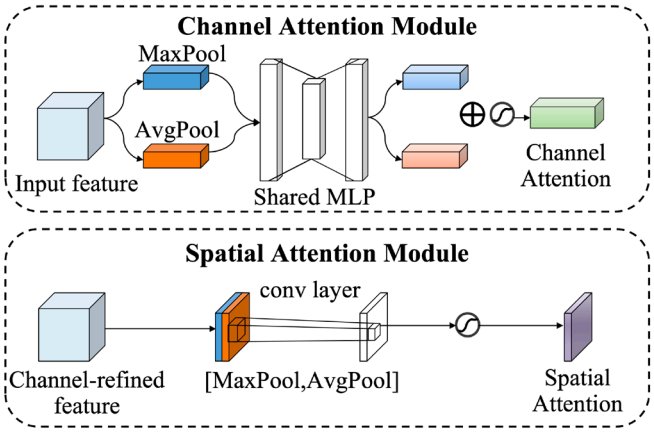


Fig. 16. CBAM structure

Our study introduces modifications to the backbone network architecture to enhance recognition performance. The detailed modified structure is provided in Figure 17.

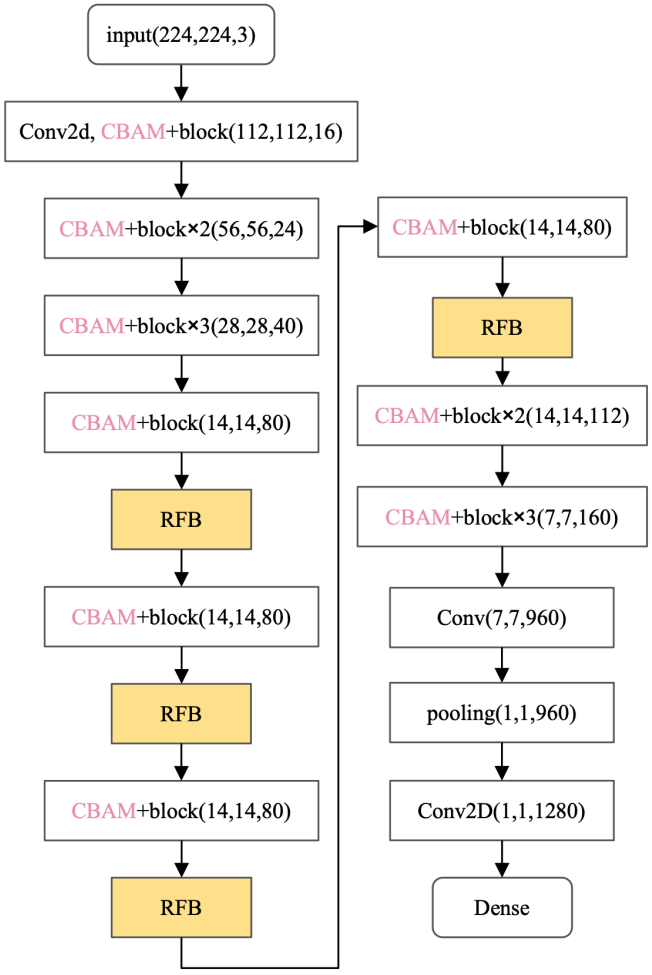


Fig. 17. RFB structure location

The quality of facial feature information significantly impacts the FaceNet backbone network's recognition results it extracts. To improve feature extraction and broaden the recognition image's receptive field, this study integrates four RFB (Receptive Field Block) structures into the FaceNet backbone. The RFB module aims to boost detection accuracy while keeping the network efficient and computationally lightweight by enhancing feature representation from a receptive field perspective. It achieves this by incorporating a dilated convolution inspired by the Inception model, effectively broadening the receptive field and maintaining generalization capabilities when used with MobileNet. The RFB's structural details are provided in Figure 18.

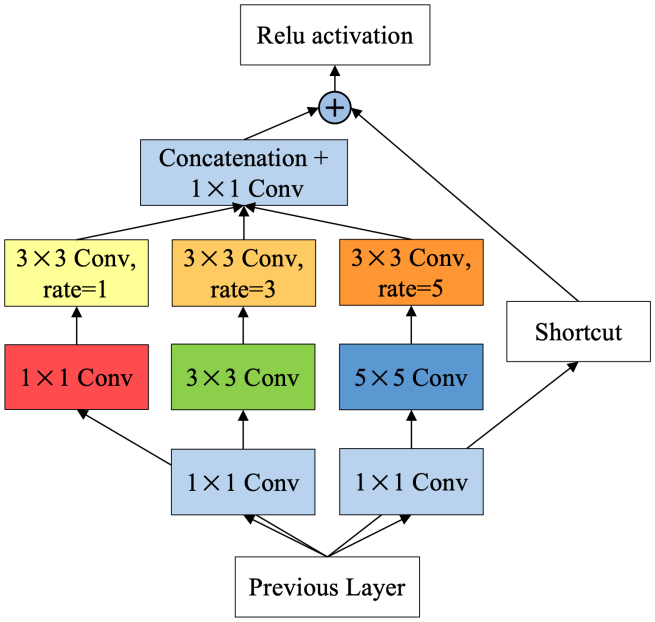


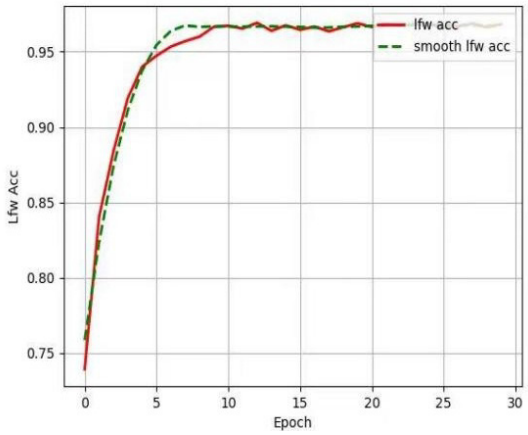
Fig. 18. RFB structure diagram

In order to compare the performance of a face recognition algorithm that integrates the CBAM attention mechanism and RFB structure, experimental validation was carried out using the CASIA-WebFace dataset for 30 epochs. The outcomes of this validation are detailed in Table VI.

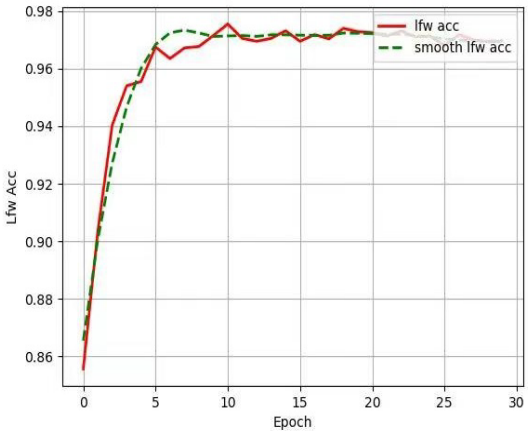
TABLE VI
EXPERIMENTAL COMPARISON OF DIFFERENT STRUCTURES

Model	Training Dataset	Number of Iterations	Accuracy
FaceNet	CASIA-WebFace	30	96.96%
FaceNet+CBAM	CASIA-WebFace	30	97.01%
FaceNet+RFB	CASIA-WebFace	30	97.12%
FaceNet+CBAM+RFB	CASIA-WebFace	30	97.36%

The training curves for the FaceNet-MN (utilizing MobileNet as the backbone) and FaceNet-MNR (employing MobileNet as the backbone with the addition of four RFB structures and the CBAM attention mechanism) models are depicted in Figure 19. The figure's red solid line represents the accuracy curve obtained during training, while the green dotted line portrays a smoothed version of these results. It is evident from the visualization that the FaceNet-MNR model attains higher accuracy, peaking at a maximum rate of 97.36%, which outperforms the original FaceNet-MN model.



a) FaceNet-MN training accuracy rate curve



b) FaceNet-MNR training accuracy rate curve

Fig. 19. Model training accuracy curve

Moreover, this paper compares the accuracy rates attained by the proposed algorithm and other methods using the LFW dataset, as illustrated in Table VII. The results manifest that the algorithm introduced in this study markedly amplifies accuracy compared to conventional face recognition algorithms, accomplishing a substantial advancement in performance.

TABLE VII
EXPERIMENTAL COMPARISON OF DIFFERENT ALGORITHMS

Algorithm	Accuracy
combined Joint Bayesian	0.9242
high-dimLBP	0.9517
CNN-3DMM estimation	0.9235
FR + FCN	0.9645
FaceNet	0.9696
DeepFace	0.9735
model in this paper	0.9736

F. Prediction Process and Effect Demonstration of FaceNet

Figure 20 illustrates the input process for the FaceNet algorithm, which begins by taking two images as input. These images first undergo a series of preprocessing steps [24]. Initially, an undistorted resize operation is applied to ensure image consistency. Subsequently, they are resized to meet the specific requirements of the FaceNet algorithm model, including normalization and dimension adjustment.

After preprocessing, the images are input into the FaceNet model, which produces feature vectors for each image. The algorithm computes the Euclidean distance between these vectors to determine similarity. If the distance is less than a specified threshold, the algorithm identifies the images as portraying the same person. Conversely, if the distance exceeds the threshold, it concludes that the images represent different individuals.

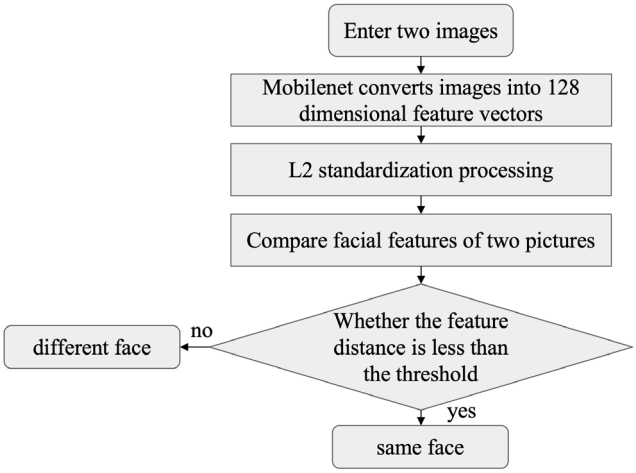
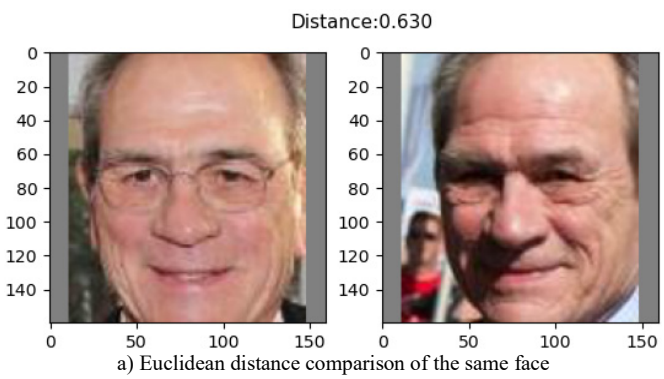
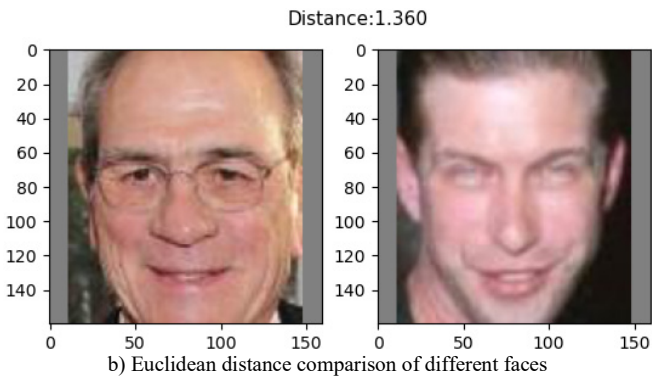


Fig. 20. Prediction process of FaceNet

The face recognition threshold is illustrated in Figure 21 and is set at 1.04. For the first image pair, the calculated distance is 0.630, which is below the threshold, indicating that the images are of the same individual. In contrast, the distance for the second image pair is 1.360, surpassing the threshold and identifying the images as depicting different individuals. These examples highlight the effectiveness of the FaceNet algorithm in discerning whether two faces correspond to the same person.



a) Euclidean distance comparison of the same face



b) Euclidean distance comparison of different faces

Fig. 21. FaceNet checks the rendering

IV. Realization of Face Recognition and Interception in Video

A. Construction of Face Recognition and Interception Network

Figure 22 depicts the process of face recognition and interception in three main stages. The initial stage utilizes the RetinaFace algorithm to detect faces in the input images. Once the face detection is complete, the system records the coordinates of the face frames within the images. These coordinates are then employed to crop and align the faces.

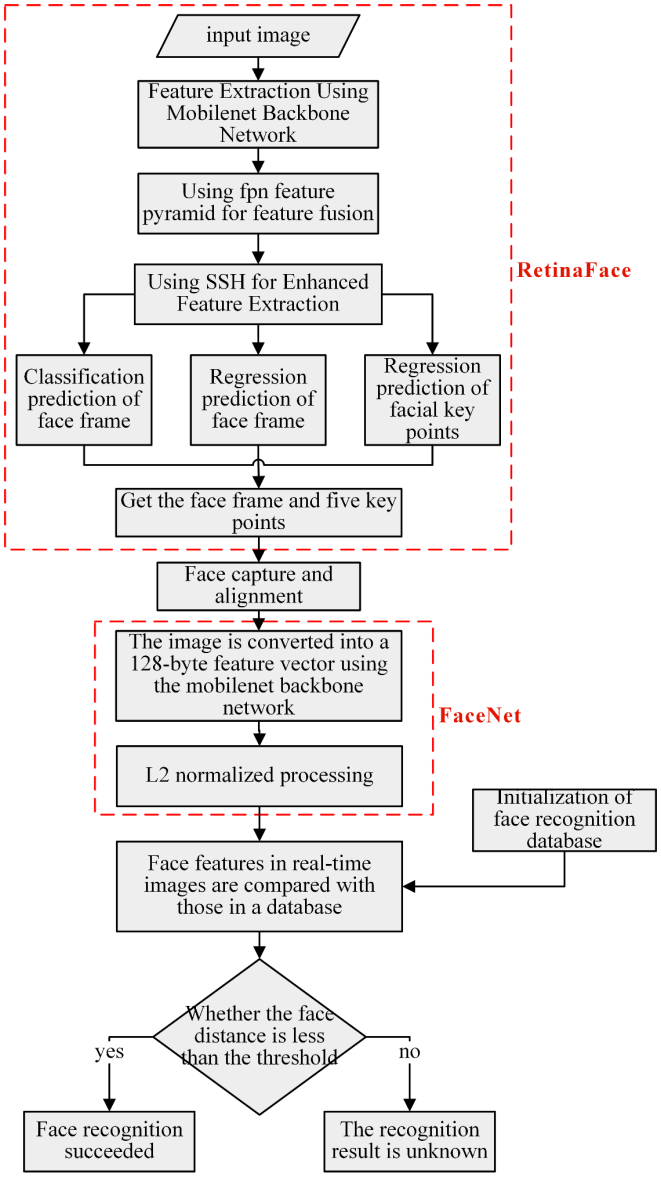


Fig. 22. The process of face recognition

In the second stage, the cropped faces undergo processing using the FaceNet algorithm, which converts them into 128-dimensional feature vectors. These vectors capture the intricate facial features extracted from the images.

The final stage involves face comparison. In this stage, the newly generated feature vectors are compared with the existing eigenvectors in the database by computing the Euclidean distance between them. This distance measures dissimilarity between each detected face and the faces stored in the database. The system then iteratively identifies the face with the shortest Euclidean distance. If this distance is below a predetermined threshold, it indicates that the detected face corresponds to a face in the database, confirming the individual's identity.

Construction And Initialization of the Face Database

Figure 23 illustrates the compilation of a face database comprising images of individuals for recognition purposes. Each image is labeled with a unique file name prefix corresponding to the person's identifier. For example, 'person1_1.jpg' denotes the first image of person1's face. It's essential to ensure that each image contains only one targeted face, and multiple images may be associated with the same individual, such as 'person1_2.jpg' and 'person1_3.jpg'. Each image in the database is mapped to a unique individual.

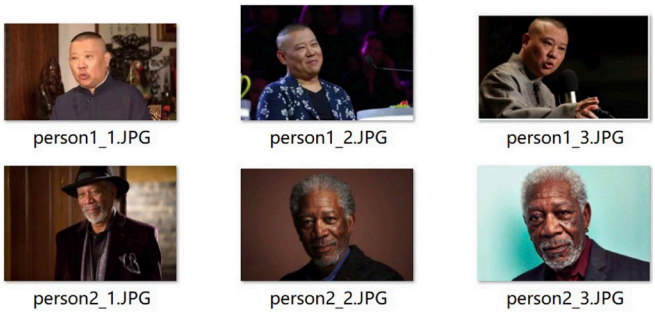


Fig. 23. Face database

The initial setup of the face database [25] involves a series of steps to prepare for face recognition by identifying and encoding the faces that the system should recognize. The initialization process includes the following steps:

(1) Traversing all images within the database.

(2) Utilizing RetinaFace to locate the face within each image.

(3) Cropping the detected face from the image.

(4) Aligning the cropped faces.

(5) Encoding each face using the FaceNet algorithm.

(6) Compiling the encoding results of all faces into a list.

(7) This list comprises the feature vector repository for all faces in the database, which will be used to establish identity by comparing with faces in real-time images.

Comparison of Target Faces in the Database with Real-time Images

The procedure for comparing real-time images with the face database is presented in Figure 24.

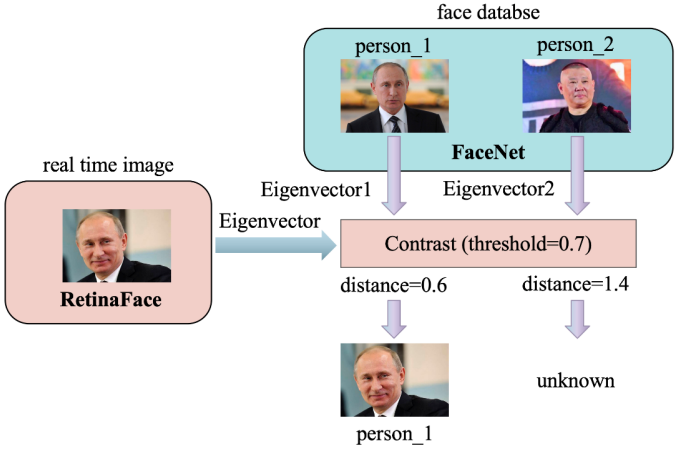


Fig. 24. Check faces against a database

In the real-time face detection process, the following steps are performed iteratively for each detected face:

(1) Retrieve the 128-dimensional feature vector from the detected face image.

(2) Compare this vector with each feature vector in the database using the Euclidean distance calculation.

(3) Identify the database face that best matches the detected face.

(4) If the calculated Euclidean distance is below a specified threshold, the system infers a successful recognition and identifies the individual.

Display of Prediction Results

The prediction results are presented in Figure 25. In Figure a), there is an image of person_1; in Figure b), person_2 is shown. Both individuals' faces are in the database, so after the network processes the images, they are framed and labeled with their names at the lower-left corner of the frame.

a) person_1 b) person_2

c) unknown d) unknown

Fig. 25. Prediction effect of face recognition network

However, in Figures c) and d), the faces do not match any in the dataset. As a result, when processed, the network frames the faces but cannot identify them, leaving the identity information unrecognized.

B. Experiment on Intercepting Clips of Target Characters in Videos

Introduction to the Experimental Environment

The setup for the experimental environment is detailed in Table VIII.

TABLE VIII
EXPERIMENTAL ENVIRONMENT

Experimental environment	configure
operating system	Windows10
CPU	Intel Core i7-10700F @ 2.90GHz
GPU	NVIDIA GeForce RTX 3070 8G
GPU acceleration library	CUDA 11.3, cudnn 7.6.5
language	Python3.8
compiler	Pycharm 2023.1
Deep learning framework	Pytorch 1.11.0

Experimental Design

The process for video capture based on face recognition is depicted in Figure 26. The workflow follows: the input video data is read frame by frame. The face detection module analyzes each frame to determine the presence of a face. If a face is detected, the face recognition module extracts facial feature information for that frame and compares it with the feature information in the face database. If a match is found, indicating the detected face corresponds to the target individual, that frame's information is recorded. This process continues for each subsequent frame until the video concludes. The collected frame information is then compiled to create a new video.

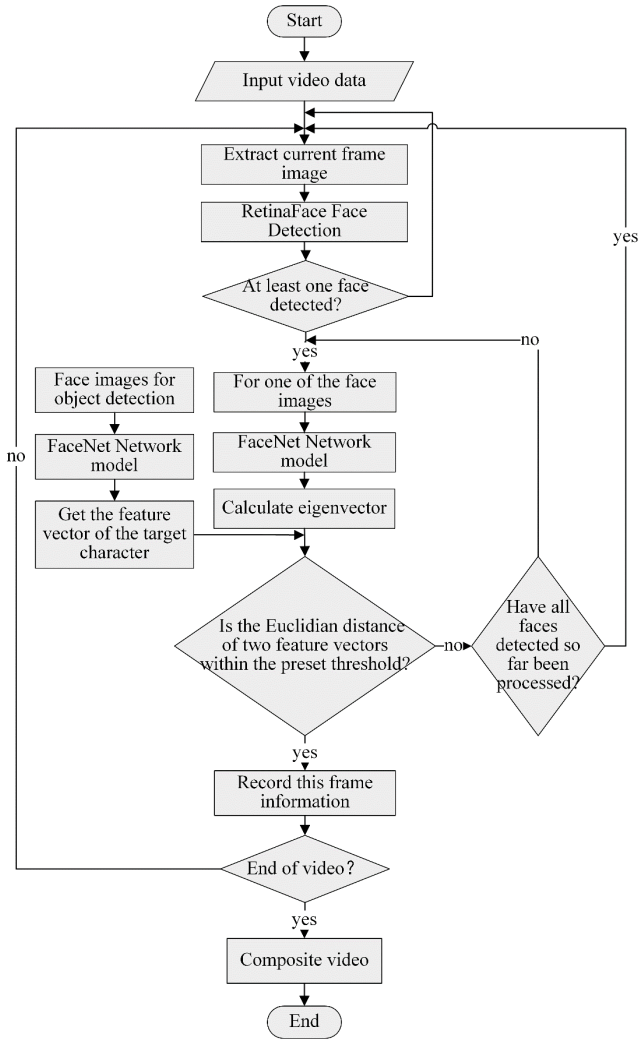


Fig. 26. Experimental flowchart

Experimental Running Test

For the test, a video file is downloaded from the internet. The face detection and recognition algorithm described in this paper is then applied to conduct a video-cutting experiment. The video used in this experiment is 59 seconds in length. The target individual's face image is entered into the face database to create the corresponding face data. The experimental process is illustrated in Figure 27.

During video face recognition. Compare the current video frame with the face database. If the current frame recognizes a face similar to the face library, the current frame is saved. Each video frame is compared to the face database during video face recognition. If a frame is recognized as containing

a face similar to one in the database, it is saved. As indicated in Figure 28, video frames are saved individually with the picture name reflecting the frame number from the video. Upon completion of the video recognition, the saved frames are used to assemble a new video, effectively capturing the segments featuring the characteristic target individual.

a) The face in the video is the same as the recognized face

b) The face in the video is different from the recognized face

Fig. 27. Experimental operation effect

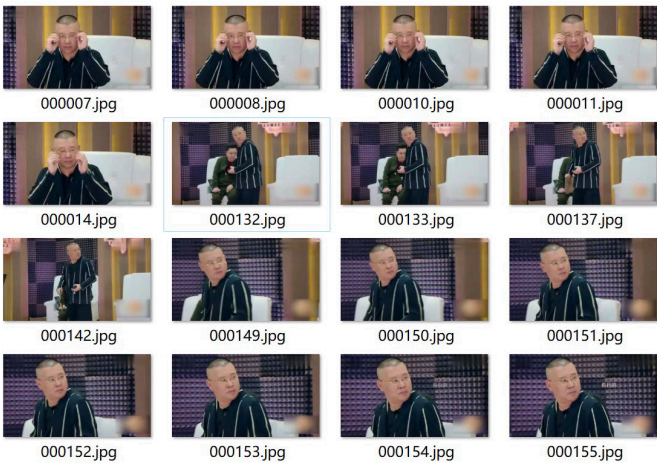


Fig. 28. Partially saved frame

Experimental Process

The research gathered ten online videos of different durations for experimental analysis. We also acquired and stored images of pertinent individuals ahead of time. These videos encompass a range of challenges, such as occlusions, facial poses, and varying face sizes.

The study evaluates the model using Precision, Recall, and F1-score metrics to gauge the enhanced algorithm's performance. As per Table IX, the version of RetinaFace known as RetinaFace (FPN) represents the improved iteration in this study. Similarly, in this paper, FaceNet (CBAM+RFB) denotes the optimized FaceNet model. Comparative analysis indicates that the precision and recall of the enhanced algorithms are 5% and 5.34% higher, respectively than those of the original versions.

TABLE IX
COMPARISON OF ABLATION EXPERIMENTS OF VARIOUS OPTIMIZATION ALGORITHMS

RetinaFace (FPN)	FaceNet (CBAM+RFB)	Precision	Recall	F1-score
		77.94%	76.55%	0.77
√		80.99%	79.66%	0.80
	√	81.55%	80.69%	0.81
√	√	83.26%	81.89%	0.82

The evaluation standard of this experiment is the accuracy rate of the interception of the target person, as shown in Equation (8).

$$v = h/s * 100\% \tag{8}$$

In Equation (8), v represents the accuracy rate of target person interception. h is the length of time that the target person appears. s represents the time the target person appears in the video segment. Aggregating data from all ten videos, the system's total interception accuracy is 79.94%.

Additionally, a separate video capture experiment was conducted using the Dlib model for face recognition, with the results detailed in Table X. When consolidating all data from the ten videos, the overall interception accuracy of the Dlib-based experiment stands at 75.55%. In comparison, the model proposed in this paper achieves a higher total interception accuracy of 79.94%.

Analysis of Experimental Results

Among the ten online videos tested, the highest face recognition-based interception accuracy was 91.30%, and the lowest was 62.96%. Comparing the interception accuracy of these videos with the original footage, it was observed that the segments with higher accuracy typically featured higher video quality, with faces visible to the camera and less complex backgrounds. In contrast, videos with lower interception accuracy often suffered from adverse lighting and excessive variation in facial positioning. The system's feasibility was further substantiated by a comparative experiment with the Dlib model using the same set of videos.

V. CONCLUSION

This study presents an enhanced automatic clipping technique for video segments featuring targeted individuals, leveraging improvements in both RetinaFace and FaceNet algorithms. The RetinaFace algorithm was refined by employing MobileNetV1-0.25 as the backbone and optimizing the Feature Pyramid Network (FPN) to exclude detections of smaller faces. Analyzing the processing speed for 460 images with a 720P resolution, the enhanced algorithm achieved a frames per second (FPS) rate 76.1, marking a 26.2% improvement over the original model's 60.3 FPS. Moreover, the detection speed outperformed those of the VGG and CNN algorithms. Additionally, the average processing time per image was reduced by 3.4 ms to 13.1 ms, culminating in an average detection speed increase of roughly 20.6%.

TABLE X
COMPARISON OF EXPERIMENTAL RESULTS WITH DLIB MODEL

Video segment	Video duration	The duration of the appearance of the target person	The duration of the model interception in this paper	The duration of Dlib model interception	The accuracy of the model interception in this paper	Accuracy of Dlib model interception
Video 1	59 s	23 s	21 s	20 s	91.30%	86.96%
Video 2	3 minutes 34 seconds	2 minutes 57 seconds	2 minutes 05 seconds	2 minutes 07 seconds	70.62%	71.75%
Video 3	2 minutes 44 seconds	1 minute 31 seconds	1 minute 15 seconds	1 minute 05 seconds	82.41%	71.43%
Video 4	5 minutes 18 seconds	2 minutes 23 seconds	1 minute 50 seconds	1 minute 50 seconds	76.92%	76.92%
Video 5	3 minutes 18 seconds	1 minute 41 seconds	1 minute 18 seconds	1 minute 17 seconds	77.22%	76.24%
Video 6	1 minute 11 seconds	54 seconds	34 seconds	34 seconds	62.96%	62.96%
Video 7	1 minute 08 seconds	42 seconds	38 seconds	36 seconds	90.48%	85.71%
Video 8	2 minutes 05 seconds	1 minute 38 seconds	1 minute 27 seconds	1 minute 27 seconds	88.78%	88.78%
Video 9	2 minutes 58 seconds	1 minute 21 seconds	1 minute 09 seconds	52 seconds	85.18%	64.20%
Video 10	55 seconds	34 seconds	25 seconds	24 seconds	73.53%	70.59%

Furthermore, FaceNet was refined by adopting MobileNetV3 as the backbone, which showed superior model size and detection accuracy. The FaceNet backbone was augmented with four Receptive Field Block (RFB) structures to broaden the receptive field, and the Convolutional Block Attention Module (CBAM) was incorporated to boost feature extraction capabilities. The enhanced model achieved an accuracy of up to 97.36%, which exceeded the original FaceNet's 96.96%.

The system's performance was tested across 10 cinematic video clips when integrating the facial detection and recognition capabilities. The combined enhancements in RetinaFace and FaceNet led to a 5% increase in precision and a 5.34% increase in recall. Compared with the Dlib model, the overall clipping accuracy of the proposed model was 79.94%, outperforming the Dlib model's 75.55% under identical conditions. The analysis of experimental results demonstrates the system's feasibility and practical applicability.

This study successfully implements an automatic clipping function for segments featuring targeted individuals in videos. Although the technology can automatically edit segments featuring targeted individuals, recognition speed may decrease in low video quality or overly complex scenes. Additionally, processing large video files remains time-consuming, highlighting significant opportunities for further improvements. Enhancing video clipping speed under these conditions remains a key focus for future research.

REFERENCES

[1] Xiuping Zhao, "The essential attributes of video data, its characteristics and its application value," *Journal of Beijing Police College*, vol. 202, no.05, pp96-101, 2022

[2] Li Zuo, "Exploring the choice of copyright protection path for online short videos," *Voice and Screen World*, vol. 2022, no.23, pp20-22, 2022

[3] Yuxin Yuan, Nong Zhang, Changliang Han, Sen Yang, Zhengzheng Xie, Jin Wang, "Digital image processing-based automatic detection algorithm of cross joint trace and its application in mining roadway excavation practice," *International Journal of Mining Science and Technology*, vol. 32, no.6, pp1219-1231, 2022

[4] Chi Jing, Zhang Haopeng, Chin Kim On, Ervin Gubin Moung, and Patricia Anthony, "Face Recognition Based on Deep Convolutional Support Vector Machine with Bottleneck Attention," *IAENG International Journal of Computer Science*, vol. 49, no.4, pp1284-1296, 2022

[5] Longlei Cui, and Ying Tian, "Facial Expression Recognition by Regional Attention and Multi-task Learning," *Engineering Letters*, vol. 29, no.3, pp919-925, 2021

[6] Chi-Hung Chuang, Cheng-Tan Tung, Yuan-Song Chang, Edward Lin, and Chih-Ping Yen, "Facial Feature Classification of Drug Addicts Using Deep Learning," *Engineering Letters*, vol. 31, no.3, pp1096-1103, 2023

[7] HongShe Dang, Guodong Di, Xuande Zhang, "Modified occlusion face detection algorithm for RetinaFace," *Experimental Technology and Management*, vol. 39, no.10, pp80-85, 2022

[8] Bintian Xue, "Design and Implementation of Improved MBLBP Based Face Detection Algorithm," *Practical Electronics*, vol. 30, no.17, pp53-56, 2022

[9] Mengru Ren, Honglu Hou, Xiulai Han, "Pedestrian Detection Based on Gabor Feature Combined with Fast HOG Feature," *Computer Systems and Applications*, vol. 30, no.10, pp259-263, 2021

[10] Huichao Fan, "Application of machine vision technology in industrial inspection," *Digital Communication World*, vol. 2020, no.12, pp156-157, 2020

[11] Hosein Arman, Abdollah Hadi-Vencheh, Reza Kiani Mavi, Mehdi Khodadadipour, Ali Jamshidi, "Revisiting the Interval and Fuzzy TOPSIS Methods: Is Euclidean Distance a Suitable Tool to Measure the Differences between Fuzzy Numbers? ," *Complexity*, vol. 2022, pp7032662, 2022

[12] "Google FaceNet scores almost 100% recognition," *Biometric Technology Today*, vol. 2015, no.4, pp2-3, 2015

[13] Matthew D Zeiler, Rob Fergus. "Visualizing and Understanding Convolutional Networks," *arXiv preprint arXiv:1311.2901*(2013)

[14] Zhenyao Zhu, Ping Luo, Xiaogang Wang, Xiaoou Tang. "Recover Canonical-View Faces in the Wild with Deep Neural Networks," *arXiv preprint arXiv:1404.3543*(2014)

[15] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," in *2014 IEEE Conference on Computer Vision and Pattern Recognitions (CVPR)*, 2014, pp1701-1708

[16] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, et al. "Going Deeper with Convolutions," *arXiv preprint arXiv:1409.4842*(2014)
[17] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna, "Rethinking the Inception Architecture for Computer Vision," *arXiv preprint arXiv:1512.00567*(2015)
[18] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, Alex Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," *arXiv preprint arXiv:1602.07261*(2016)
[19] Wenyi Hu, Hongkun Wang, Yujia Du, "Identification Method of Tomato Diseases and Pests Based on SE Module and ResNet," *Agriculture Engineering*, vol. 12, no.9, pp33-40, 2022
[20] Shuo Yang, Ping Luo, Chen Change Loy, Xiaoou Tang, "WIDER FACE: A Face Detection Benchmark," *arXiv preprint arXiv:1511.06523*(2015)
[21] Preetha S, Sheela S V, "Security Monitoring System Using FaceNet For Wireless Sensor Network," *arXiv preprint arXiv:2112.01305*(2021)
[22] Bolun Ding，Guangmei Fang，Shude Liu, "A combined forecasting model of Chinese pig price index based on L2 norm GOWA operator," *Journal of Ningxia Teachers University*, vol. 43, no.1, pp69-75, 2022
[23] Yuliang Gao, Xiangying Xu, Yonglong Zhang, Yifeng Gu, Lifeng Zhang,Bin LI, "Image recognition algorithm of rice diseases and insect pests based on shuffle attention mechanism," *Journal of Yangzhou University(Natural Science Edition)*, vol. 24, no.6, pp53-57, 2021
[24] Li Jiang, Zhemin Sun, Hanlin Jiang, Tianwei Cai, "Preprocessing Method of License Plate Recognition," *Computer Knowledge and Technology*, vol. 16, no.25, pp178-179, 2020
[25] Z. Huang, S. Shan, R. Wang, H. Zhang, S. Lao, A. Kuerban, et al., "A Benchmark and Comparative Study of Video-Based Face Recognition on COX Face Database," *IEEE Transactions on Image Processing*, vol. 24, no.12, pp5967-5981, 2015